\* SIMULATION\_CODE\_FEW\_STUDIES.SAS

This program performs a simulation study for assessing the beta-binomial regression model with a low number of studies.

It introduces the SAS macro %FewStudiesSim which performs the simulation (data generation, computation of starting values, parameter estimation, and summarization

of results in a single data set) for a fixed simulation scenario. A single simulation scenario is defined by the macro parameters

true\_or : Treatment effect

rem : =1 Generate data from the standard random effects model

=0 Generate data from the standard fixed effects model

large\_study : =1 generates one large study (first study of each simrun 10 times larger)

=0 (Default) all studies are completly simulated accroding the distribution of Turner et al.

nsimruns : Number of generated meta-analyses

nstudy : Number of studies in a single meta-analysis

seedsim : Initializing value of the random number generator

printsimdata : =1 Print the simulated data

=0 (Default) Do not print the simulated data

options pagesize=**60** linesize=**120**;

libname result "C:\Users\tmathes\temp\Simulation";

**%macro** FewStudiesSim(nstudy=,true\_pc=,true\_pt=,true\_or=,rem=,large\_study=**0**,seedsim=,nsimruns=,printsimdata=**0**);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* PRELIMINARIES \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

options nomprint nomlogic source notes;

proc printto;

options nomlogic nomprint nosource nonotes nosymbolgen;

proc printto log="C:\Users\tmathes\temp\Simulation\Log\_&seedsim..txt"

print="C:\Users\tmathes\temp\Simulation\Output\_&seedsim..txt";

run;

proc format;

value treatf **1**=" Treatment" **0**="Control";

value eventf **1**=" Yes" **0**="No";

value $ident\_t 'H0'='No treatment effect' 'H1'='Medium treatment effect';

run;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\* DATA GENERATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* The data set SIMDATA contains the simulated data sets, the variable SIMRUNS indexes the single simulation run;

data simdata;

\* Initialize random seed;

call streaminit(&seedsim);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Loop across the number of simulation runs \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

do simruns=**1** to &nsimruns;

\*\*\*\*\* Begin data generation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* ... Declare number of studies from the macro variable &nstudy;

NumberofStudies=&nstudy;

\* Generate a baseline event probability from a beta distribution with Alpha=0.393628 und Beta=0.881246.

These values originate from an analysis of the 14,886 meta-analyses with binary outcomes from Rebecca Turner

(vgl. Analyse\_Baseline\_Risks\_Turner.SAS)

A decent fit results from fitting a beta distribution with alpha=0.422932 and beta=1.433449. To be concrete,

comparing observed (from Turner) vs. fitted summary statistics, we find for the

Median 0.1262 vs. 0.1325, for the

Mean 0.2271 vs. 0.2336 and for the

standard deviation 0.2558 vs. 0.2556

The true mean and standard deviation from a Beta(0.422932, 1.433449) distribution are 0.22783 and 0.24817

ATTENTION: The fact that the baseline probabilities are simulated from a beta distribution DOES NOT MEAN that the

data are generated from the beta-binomial model. We here generate one fixed baseline probability for a single meta-analysis,

whereas the beta-binomial model assumes that the baseline probabilities OF THE STUDIES WITHIN THE META-ANALYSIS follow

a beta distribution.

The true model is still a standard inverse-variance random-effects model;

true\_pc=rand('BETA',**0.422932**,**1.433449**);

true\_logit\_pc=log(true\_pc/(**1**-true\_pc));

true\_OR=&true\_OR;

true\_LogOR=log(true\_OR);

\* For the random effects situation, a taus-square has to be generated;

%if &rem=**1** %then %do;

\* Generation of the random effects variance (tau-square);

LogNormal\_mean\_Turner\_Tau=-**1.47**;LogNormal\_stddev\_Turner\_Tau=**1.65**;

\* Generation of a skewed distribution using Fleishmans power transformation method

(Fleishman AI. A method for simulating non-normal distributions. Psychometrika, 1978, 43, 521-532) as described in Fan et al.

(Fan X, Felsovalyi A, Sivo SA, Kennan SC. SAS for Monte Carlo Studies: A Guide for Quantitative Researchers. 2002 (Chapter 4.2.2.2)) ;

random\_standard\_normal=rand("NORMAL");

random\_skewed\_normal = **0.104796973** + **1.049992872**\*random\_standard\_normal

-**0.104796973**\*random\_standard\_normal\*\***2** - **0.020782441**\*random\_standard\_normal\*\***3**;

Log\_TauSquare\_skewed=LogNormal\_mean\_Turner\_Tau + LogNormal\_stddev\_Turner\_Tau\*random\_skewed\_normal;

true\_TauSquare=exp(Log\_TauSquare\_skewed);

%end;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Loop across the number of studies within a meta-analysis \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

do study=**1** to NumberofStudies;

\* Generate the sample size for the two groups following the distribution in Turner et al., Table 2;

LogNormal\_mean\_Turner\_SSize=**4.615**;LogNormal\_stddev\_Turner\_SSize=**1.1**;

Log\_StudySize=LogNormal\_mean\_Turner\_SSize + LogNormal\_stddev\_Turner\_SSize\*rand('NORMAL');

\* Determine the study size by doubling the "ceiled" group size, where ceiling means

calculating the smallest integer that is greater than or equal to the argument;

\* In the homogeneous study size scenario all studies follow the distribution in Turner et al.;

%if &large\_study=**0** %then %do;

StudySize=**1**+ceil(exp(Log\_StudySize));

%end;

\* in In the inhomogeneous study size scenario the number of studies in the meta-analysis minus one follow the distribution in Turner et al, and one

study is ten times larger as expected the distribution in Turner et al.;

%if &large\_study=**1** %then %do;

if study=**1** then do; StudySize=**1**+ceil(exp(Log\_StudySize))\***10**; end;

else do; StudySize=**1**+ceil(exp(Log\_StudySize));end;

%end;

\* In the random effects situation, the true event probabilities are varying according to the random Log Odds Ratio;

%if &rem=**1** %then %do;

true\_TauSquare\_i=true\_TauSquare\*rand('NORMAL');

\* Calculate the study-specific event probability in the treatment group on the expit scale.

The linear predictor here is the sum of the logit of the event probability in the control group, the true LogOR und the random tau-square in the

respective study;

true\_pt\_i=exp(true\_logit\_pc + true\_LogOR + true\_TauSquare\_i)/(**1**+exp(true\_logit\_pc + true\_LogOR + true\_TauSquare\_i));

%end;

%if &rem=**0** %then %do;

true\_pt=exp(true\_logit\_pc + true\_LogOR)/(**1**+exp(true\_logit\_pc + true\_LogOR));

%end;

\* Simulate the sample size in the treatment group by mimicking a randomization process with

a binomial probability 0.5;

nt=rand('BINOMIAL',**0.5**,studysize);

nc=studysize-nt;

\* Exclude extrem cases with zero treated observations in a treatment group.

ATTENTION: These statement does not exclude zero events in the groups! (st=0 or sc=0);

if nt=**0** then do; nt=**1**; nc=nc-**1**;end;

if nc=**0** then do; nc=**1**; nt=nt-**1**;end;

\* Generate the number of events in both groups;

sc=rand('BINOMIAL',true\_pc,nc);

%if &rem=**0** %then %do;st=rand('BINOMIAL',true\_pt,nt); %end;

%if &rem=**1** %then %do;st=rand('BINOMIAL',true\_pt\_i,nt); %end;

\*\*\*\*\* End of data generation \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Estimate raw event probabilities;

pt=st/nt; pc=sc/nc;

nsucc=st+sc;

ntrials=nt+nc;

p\_overall=nsucc/ntrials;

test\_OR=(pt\*(**1**-pc))/(pc\*(**1**-pt));

test\_LogOR=log(test\_OR);

weight\_test\_LogOR=**1**/(**1**/st + **1**/(nt-st) + **1**/sc + **1**/(nc-sc));

\* Check for double-zero and single-zero trials;

/\*

doublezero=0; if (sc=0 and st=0) or (nt=st and nc=sc) then doublezero=1;

singlezero\_control=0; if sc=0 and st ne 0 then singlezero\_control=1;

singlezero\_treatment=0; if st=0 and sc ne 0 then singlezero\_treatment=1;

\*/

\* Calculate logits of the estimated event probabilities (which are needed for the generation of starting values)

In groups without an event the logit is assigned a negative random number between -10 and -15, in groups

with only events the logit is assigned a positive random number between 5 and 10;

if pt=**0** then do; logit\_pt=-**10** - **5**\*rand('UNIFORM'); end;

if pt=**1** then do; logit\_pt= **5** + **5**\*rand('UNIFORM'); end;

if **0**<pt<**1** then do; logit\_pt=log(pt/(**1**-pt)); end;

if pc=**0** then do; logit\_pc=-**10** - **5**\*rand('UNIFORM'); end;

if pc=**1** then do; logit\_pc= **5** + **5**\*rand('UNIFORM'); end;

if **0**<pc<**1** then do; logit\_pc=log(pc/(**1**-pc)); end;

\* Calculate logarithms (not logits) of the estimated event probabilities (which are also needed for the generation of starting values)

In groups without an event the logit is assigned a negative random number between -10 and -15;

if pt=**0** or pc=**0** then do; log\_pt=-**10** - **5**\*rand('UNIFORM'); log\_pc=-**10** - **5**\*rand('UNIFORM'); end;

else do; log\_pt=log(pt); log\_pc=log(pc); end;

\* Generate a pseudo intercept for each study;

int=**1**;

output;

end; \* of loop across studies within a meta-analysis;

end; \* of loop across the number of simulation runs;

run;

%if &printsimdata=**1** %then %do;

proc print data=simdata;

\*var simruns study st sc nt nc true\_pt true\_pc true\_LogOR true\_RandomLogOR true\_OR true\_RandomOR test\_or weight\_test\_or;

title"simdata";

run;

%end;

\* Control data generation;

\* ods select none;

proc means data=simdata n mean std median min q1 q3 max;

%if &rem=**1** %then %do; var nt nc StudySize st sc pt pc test\_OR true\_pc true\_OR true\_LogOR true\_TauSquare\_i true\_pt\_i; %end;

%if &rem=**0** %then %do; var nt nc StudySize st sc pt pc test\_OR true\_pc true\_OR true\_LogOR; %end;

title"Control generation of random effects data";

run;

\* ods select all;

\* Define a id (IDENT) for the simulation scenarios;

data ident;

%if &true\_or = **1** %then %do;ident\_t='H0';%end;

%if &true\_or = **0.684** %then %do;ident\_t='H1';%end;

%if &rem = **0** %then %do;ident\_REM='FEM';%end;

%if &rem = **1** %then %do;ident\_REM='REM';%end;

%if &large\_study = **0** %then %do;ident\_largestudy='E';%end;

%if &large\_study = **1** %then %do;ident\_largestudy='L';%end;

%if &nstudy=**2** %then %do; ident\_nstudy='02';%end;

%if &nstudy=**3** %then %do; ident\_nstudy='03';%end;

%if &nstudy=**4** %then %do; ident\_nstudy='04';%end;

%if &nstudy=**5** %then %do; ident\_nstudy='05';%end;

%if &nstudy=**10** %then %do; ident\_nstudy='10';%end;

%if &nstudy=**15** %then %do; ident\_nstudy='15';%end;

%if &nstudy=**20** %then %do; ident\_nstudy='20';%end;

%if &nstudy=**30** %then %do; ident\_nstudy='30';%end;

%if &nstudy=**50** %then %do; ident\_nstudy='50';%end;

Ident = ident\_t || ident\_REM || ident\_largestudy || ident\_nstudy;

call symput("Ident",Ident);

run;

%put Ident: &Ident;

/\*

proc print data=ident noobs;

var ident ident\_t ident\_REM ident\_nstudy;

title'ident';

run;

\*/

\* Reorganize SIMDATA, so that each single study has two observations (one for treatment, one for control, SIMDATADOUBLE) or

each individual constitutes a single observation (SIMDATAEXPLODED);

data simdatadouble\_temp1;

set simdata(keep=simruns study st sc nt nc);

by simruns;

do k=**1** to **2**;output;end;

run;

data simdatadouble(keep=simruns study treatment control success n);

set simdatadouble\_temp1;

by simruns;

if mod(k,**2**)=**1** then do; treatment=**1**; control=**0**; success=st; n=nt; end;

if mod(k,**2**)=**0** then do; treatment=**0**; control=**1**; success=sc; n=nc; end;

run;

/\*proc print data=simdatadouble;title"simdatadouble";run;\*/

data simdataexplode(drop=l);

set simdatadouble;

by simruns;

do l=**1** to n;

if l <= success then event=**1**;

if l > success then event=**0**;

status=**2**-event;

output;

end;

run;

/\*proc print data=simdataexplode;title"simdataexplode";run;\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* STARTING VALUES \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Compute means, standard deviations for the estimated event probabilities in both treatment groups;

ods select none;

proc corr data=simdata spearman kendall pearson cov;

var pt;with pc;

by simruns;

ods output SimpleStats=SimpleStats\_pt(keep=Simruns Variable Mean StdDev where=(Variable="pt") rename=(mean=start\_pt StdDev=start\_stddev\_pt))

SimpleStats=SimpleStats\_pc(keep=Simruns Variable Mean StdDev where=(Variable="pc") rename=(mean=start\_pc StdDev=start\_stddev\_pc))

SpearmanCorr=SpearmanCorr(rename=(pt=start\_SpearmansRho))

KendallCorr=KendallCorr(rename=(pt=start\_KendallsTau))

PearsonCorr=PearsonCorr(rename=(pt=start\_Pearson));

run;

ods select all;

/\*proc print data=SimpleStats\_pt;run;

proc print data=SimpleStats\_pc;run;

proc print data=SpearmanCorr;run;

proc print data=KendallCorr;run;

proc print data=PearsonCorr;run;

\*/

\* Compute means, standard deviations for the logarithms for estimated event probabilities in both treatment groups;

ods select none;

proc corr data=simdata pearson cov;

var log\_pt;with log\_pc;

by simruns;

ods output SimpleStats=SimpleStats\_log\_pt\_temp(keep=Variable simruns Mean StdDev where=(Variable="log\_pt") rename=(mean=start\_logpt StdDev=start\_stddev\_logpt))

SimpleStats=SimpleStats\_log\_pc\_temp(keep=Variable simruns Mean StdDev where=(Variable="log\_pc") rename=(mean=start\_logpc StdDev=start\_stddev\_logpc))

Cov=Cov\_Logit\_temp

;

run;

ods select all;

data SimpleStats\_log\_pt; set SimpleStats\_log\_pt\_temp(drop=Variable); start\_variance\_logpt=start\_stddev\_logpt\*\***2**; run;

/\*proc print data=SimpleStats\_log\_pt;run;\*/

data SimpleStats\_log\_pc; set SimpleStats\_log\_pc\_temp(drop=Variable); start\_variance\_logpc=start\_stddev\_logpc\*\***2**; run;

/\*proc print data=SimpleStats\_log\_pc;run;\*/

data Cov\_Logit(keep=simruns Covariance\_Logs\_pt\_pc); set Cov\_Logit\_temp(drop=Variable); Covariance\_Logs\_pt\_pc=log\_pt; run;

/\*proc print data=Cov\_Logit;run;\*/

\* Starting values for the Beta-Binomial-Model;

data startingvalues\_Beta\_Binomial(keep=simruns rho b0 b\_treat);

merge SimpleStats\_pt(keep=simruns start\_pt start\_stddev\_pt)

SimpleStats\_pc(keep=simruns start\_pc start\_stddev\_pc);

\* Pooling variances and raw probabilities;

if start\_stddev\_pt=**0** then start\_stddev\_pt=**0.001**;

if start\_stddev\_pc=**0** then start\_stddev\_pc=**0.001**;

if start\_pt=**0** then start\_pt=**0.001**;

if start\_pc=**0** then start\_pc=**0.001**;

pooledvar=(start\_stddev\_pt\*\***2** + start\_stddev\_pc\*\***2**)/**2**;

pooledp =(start\_pt+start\_pc)/**2**;

rho=pooledvar/(pooledp\*(**1**-pooledp))/(**1**-pooledvar/(pooledp\*(**1**-pooledp))) ;

b0 =log(start\_pc/(**1**-start\_pc));

if (start\_pt=**0**) or (start\_pc=**0**) then do; b\_treat=**0**; end;

else do; b\_treat=log((start\_pt\*(**1**-start\_pt))/(start\_pc\*(**1**-start\_pc))); end;

by simruns;

run;

/\*proc print data=startingvalues\_Beta\_Binomial;run;\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\* ANALYSIS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\* Naming conventions:

Each procedure is named by 4 characters

MHFE : Mantel-Haenszel Fixed-Effect-Method

YPET : Yusuf-Peto-Method

DLRE : Inverse-Variance Random Effects Meta-analysis nach DerSimonian/Laird Method

HKSJ : Inverse-Variance Random Effects Meta-analysis nach Hartung/Knapp und Sidik/Jonkman Method

mKH\_ : Inverse-Variance Random Effects Meta-analysis nach modified Hartung/Knapp und Sidik/Jonkman Method

PMRE : Inverse-Variance Random Effects Meta-analysis mit Paule und Mandel heterogenity variance estimator

PMKH : Inverse-Variance Random Effects Meta-analysis mit Paule und Mandel heterogenity variance estimator und modified Hartung/Knapp Konfidenzintervall

BBIN : Beta-binomial regression

BB1N : Beta-binomial regression mit Berechnung des Konfidenzintervalls mit der Anzahl der Freiheitsgrade gleich der Anzahl der Studien

BB2N : Beta-binomial regression mit Berechnung des Konfidenzintervalls mit der Anzahl der Freiheitsgrade, wie sie von PROC NLMIXED vorgeschlagen wird,

nämlich Anzahl der Studie mal zwei

Treatment effect is named by 2 capital letters:

OR: Odds ratio

;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* EFFECT ESTIMATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* MH analysis (PROC FREQ) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*calculate the stratified Odds-Ratio and 95%-CIs according to Cochran-Mantel-Heasel;

ods listing close;

proc freq data=simdataexplode order=formatted;

tables study\*treatment\*event / cmh;

by simruns;

ods output CommonRelRisks=OR\_MaHa\_temp(where=(StudyType="Fall-Kontroll")

rename=(Value=OR\_MaHa LowerCL=CI95L\_OR\_MaHa UpperCL=CI95U\_OR\_MaHa))

;

format treatment treatf. event eventf.;

title"MH analysis, PROC FREQ";

run;

ods listing;

\*calculate the stratified Log Odds-Ratio with 95%-CIs and the standard error of the stratified Log Odds-Ratio;

data LogOR\_MaHa(keep=simruns LogOR\_MaHa CI95L\_LogOR\_MaHa CI95U\_LogOR\_MaHa SE\_LogOR\_MaHa);

set OR\_MaHa\_temp;

LogOR\_MaHa=log(OR\_MaHa); CI95L\_LogOR\_MaHa=log(CI95L\_OR\_MaHa); CI95U\_LogOR\_MaHa=log(CI95U\_OR\_MaHa);

Length\_CI95\_OR\_MaHa= CI95U\_LogOR\_MaHa-CI95L\_LogOR\_MaHa;

SE\_LogOR\_MaHa=Length\_CI95\_OR\_MaHa/(**2**\*probit(**0.975**));

run;

/\*proc print data=LogOR\_MaHa;run;\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* Yusuf-Peto (with and without Sato correction) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data yusufpeto(keep=simruns LogOR\_YPET SE\_LogOR\_YPET CI95L\_LogOR\_YPET CI95U\_LogOR\_YPET);

set simdata end=lastrecord;

by simruns;

\* Initialize terms;

if first.simruns then do;

sum\_YPET\_den=**0**; sum\_YPET\_num=**0**;

end;

\* Calculate Terms that are summed across studies ...;

YPET\_den = Nt\*Nc\*(St+Sc)\*(Nt+Nc-(St+Sc))/(((Nt+Nc)\*\***2**)\*(Nt+Nc-**1**));

YPET\_num = (St-(Nt\*(St+Sc)/(Nt+Nc)));

\* ... and sum them;

sum\_YPET\_den+YPET\_den;

sum\_YPET\_num+YPET\_num;

\* When the last observation is reached calculate estimates and output them;

if last.simruns then do;

\* Standard method;

LogOR\_YPET = sum\_YPET\_num/sum\_YPET\_den;

CI95L\_LogOR\_YPET = LogOR\_YPET - (probit(**0.975**)/sqrt(sum\_YPET\_den));

CI95U\_LogOR\_YPET = LogOR\_YPET + (probit(**0.975**)/sqrt(sum\_YPET\_den));

Length\_CI95\_OR\_YPET= CI95U\_LogOR\_YPET-CI95L\_LogOR\_YPET;

SE\_LogOR\_YPET=Length\_CI95\_OR\_YPET/(**2**\*probit(**0.975**));

\* OR\_YPET= exp(LogOR\_YPET);

\* CI95L\_OR\_YPET= exp(CI95L\_LogOR\_YPET);

\* CI95U\_OR\_YPET= exp(CI95U\_LogOR\_YPET);

output;

end;

title"Yusuf-Peto method";

run;

/\*proc print data=yusufpeto;run;\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* Inverse-Variance Random Effects Meta-analysis DerSimonian/Laird Heterogenetiy Variance estimator (DL, HKSJ, mHK)\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data ivre;

set simdata;

by simruns;

wsquare\_test\_LogOR=weight\_test\_LogOR\*weight\_test\_LogOR;

vari=**1**/weight\_test\_LogOR;

run;

\* Compute the heterogeneity statistic Q and the heterogeneity estimator Tau-square. Use the idea of

Whithead to consider Q as the residual sum of square from a weighted least sqaures regression

(Whitehead A. Meta-Analysis Of Controlled Clinical Trials. 2002, John Wiley & Sons, Ltd);

ods listing close;

proc glm data=ivre outstat=outglm(keep=ss \_source\_ simruns);

model test\_LogOR=int / inverse clparm noint;

weight weight\_test\_LogOR;

by simruns;

title"Estimate Q by PROC GLM";

run;

/\*proc print data=outglm;run;\*/

ods listing;

\* Determine the number of studies, the sum of weights and the sum of squared weights;

proc means data=ivre n;

var weight\_test\_LogOR wsquare\_test\_LogOR;

by simruns;

output out=outmeans n(true\_OR)=nstudies sum(weight\_test\_LogOR)=sumw sum(wsquare\_test\_LogOR)=sumwsquare;

run;

/\*proc print data=outmeans;run;\*/

\* Compute Q and Tau-square;

data qcalcivre(keep=simruns nstudies tau\_square Q);

merge outmeans outglm(where=(\_SOURCE\_="ERROR"));

by simruns;

tau\_square=max(**0**,(SS-(nstudies-**1**))/(sumw-sumwsquare/sumw));

isquare=(SS-((nstudies-**1**)))/SS;

Q=SS;

if isquare<**0** then isquare=**0**;

title"Compute Q and Tau-square";

run;

/\*proc print data=qcalcivre;run;\*/

\* ... explode the data set qcalc ...;

data qcalcivreexplode; set qcalcivre; do study=**1** to nstudies; output; end; run;

/\*proc print data=qcalcivreexplode;run;\*/

\* ... and merge with the original data set;

data remeins;

merge ivre qcalcivreexplode;

by simruns study;

\* Compute the random effects weight for each study;

rem\_w=**1**/(vari+tau\_square);

run;

/\*proc print data=remeins;run;\*/

\* Compute the standard random effects estimator theta\_rem and its variance;

data rem(keep=simruns nstudies LogOR\_IVRE var\_theta\_rem se\_theta\_rem);

set remeins end=lastrecord;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_rem\_times\_w=**0**;sum\_rem\_w=**0**;

end;

\* Calculate terms that are summed across studies ...;

test\_LogOR\_times\_rem\_w = rem\_w\*test\_LogOR;

\* ... and sum them;

sum\_rem\_w+rem\_w;

sum\_rem\_times\_w + test\_LogOR\_times\_rem\_w;

\* When the last observation is reached, calculate estimates and output them;

if last.simruns then do;

LogOR\_IVRE=sum\_rem\_times\_w/sum\_rem\_w;

var\_theta\_rem=**1**/sum\_rem\_w;

se\_theta\_rem=**1**/sqrt(sum\_rem\_w);

output;

end;

title"rem";

run;

/\*proc print data=rem;run;\*/

\* ... explode the data set rem ...;

data remexplode; set rem; by simruns; do study=**1** to nstudies; output; end; run;

/\*proc print data=remexplode;run;\*/

\* ... and merge with the data set remeins;

data dlhksj;

merge remeins remexplode;

by simruns study;

run;

/\* proc print data=dlhksj;run;\*/

\*\*\*Compute the estimator of DerSimonian and Laird\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data LogOR\_DLRE(keep=simruns nstudies tau\_square Q LogOR\_DLRE var\_theta\_rem se\_theta\_rem var\_DLRE CI95L\_LogOR\_DLRE CI95U\_LogOR\_DLRE se\_DLRE);

set dlhksj end=lastrecord;

LogOR\_DLRE=LogOR\_IVRE;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_rem\_w=**0**;

end;

\* ... and sum weigths;

sum\_rem\_w+rem\_w;

\* When the last observation is reached calculate estimates and output them;

if last.simruns then do;

var\_DLRE=**1**/sum\_rem\_w;

se\_DLRE=sqrt(var\_DLRE);

CI95L\_LogOR\_DLRE = LogOR\_IVRE - quantile('normal',**.975**)\* se\_DLRE;

CI95U\_LogOR\_DLRE = LogOR\_IVRE + quantile('normal',**.975**)\* se\_DLRE;

output;

end;

title"DerSimonian and Laird";

run;

/\*proc print data=LogOR\_DLRE; run;\*/

\*\*\*Compute the estimator of Hartung/Knapp or Sidik/Jonkma\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

data LogOR\_HKSJ(keep=simruns nstudies tau\_square Q LogOR\_HKSJ LogOR\_mKH\_ var\_theta\_rem se\_theta\_rem var\_hksj

se\_hksj CI95L\_LogOR\_HKSJ CI95U\_LogOR\_HKSJ q\_mKH\_ var\_mKH\_ se\_mKH\_ CI95L\_LogOR\_mKH\_ CI95U\_LogOR\_mKH\_);

set dlhksj end=lastrecord;

LogOR\_HKSJ=LogOR\_IVRE;

LogOR\_mKH\_=LogOR\_IVRE;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_num\_hksj=**0**;

sum\_rem\_w=**0**;

end;

\* Calculate Terms that are summed across studies ...;

num\_hksj = rem\_w\*(test\_LogOR-LogOR\_IVRE)\*\***2**;

\* ... and sum them;

sum\_num\_hksj+num\_hksj;

sum\_rem\_w+rem\_w;

\* When the last observation is reached calculate estimates and output them;

if last.simruns then do;

var\_hksj=sum\_num\_hksj/((nstudies-**1**)\*sum\_rem\_w);

se\_hksj=sqrt(var\_hksj);

q\_mKH\_=max(**1**,sum\_num\_hksj/(nstudies-**1**));

var\_mKH\_=q\_mKH\_/sum\_rem\_w;

se\_mKH\_=sqrt(var\_mKH\_);

CI95L\_LogOR\_HKSJ = LogOR\_HKSJ - quantile('T',**.975**,nstudies-**1**)\* se\_hksj;

CI95U\_LogOR\_HKSJ = LogOR\_HKSJ + quantile('T',**.975**,nstudies-**1**)\* se\_hksj;

CI95L\_LogOR\_mKH\_ = LogOR\_mKH\_ - quantile('T',**.975**,nstudies-**1**)\* se\_mKH\_;

CI95U\_LogOR\_mKH\_ = LogOR\_mKH\_ + quantile('T',**.975**,nstudies-**1**)\* se\_mKH\_;

output;

end;

title"Hartung/Knapp, Sidik/Jonkman";

/\*proc print data=LogOR\_HKSJ; run;

run;\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* Inverse-Variance Random Effects Meta-analysis Paule/Mandel Heterogenity Variance Estimator (PM, PMHK)\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*prepare dataset for Paule and Mandel Macro;

data pmivre;

set simdata;

vari=**1**/weight\_test\_LogOR;

run;

\* Determine the number of studies;

proc means data=pmivre n;

var study;

by simruns;

output out=outmeans n(study)=nstudies;

run;

/\*proc print data=outmeans;\*/

\* ... and merge with the original data set;

data pmstart (keep=simruns nstudies study test\_LogOR vari);

merge outmeans pmivre;

by simruns;

run;

/\*proc print data=pmstart;\*/

\*macro to estimate the Paule and Mandel heterogenity variance etimator Tau(pm);

**%macro** PmEstimator(data=, start\_tau\_square\_pm=, nstudies=, iterations=);

data pm;

set &data;

if tau\_square\_pm=**.** then tau\_square\_pm=&start\_tau\_square\_pm;

%let max\_fin\_f\_tau=.;

%let \_i=0;

%put &max\_fin\_f\_tau;

%put &\_i;

\*starting the loop;

%do %until(&max\_fin\_f\_tau<=**0** or &\_i=&iterations);

%let \_i = %eval(&\_i+1);

run;

/\*proc print data=pm;\*/

\*calculate weigths and terms that are necessary to calculate the Paule and Mandel estimating equation;

data pm;

set pm;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_w\_pm=**0**; sum\_num\_yw\_tau=**0**;

end;

\* Calculate terms that are summed across studies ...;

w\_pm=**1**/(tau\_square\_pm+vari);

num\_yw\_tau=w\_pm\*test\_LogOR;

sum\_w\_pm+w\_pm;

sum\_num\_yw\_tau+num\_yw\_tau;

\* and sum them...;

if last.simruns then do;

yw\_tau=sum\_num\_yw\_tau/sum\_w\_pm;

end;

run;

/\*proc print data=pm;run;\*/

\*imput missing column values necessary for further summed terms...;

data pm;

do until(yw\_tau);

set pm end=last;

if not missing(yw\_tau) then complet\_yw\_tau=yw\_tau;

end;

do until(yw\_tau);

set pm end=\_last;

output;

end;

run;

/\*proc print data=pmd;\*/

\*calulate the Paule and Mandel estimating equation (f\_tau), the target equation (fin\_f\_tau) and the correction (delta\_tau);

data pm;

set pm;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_f\_tau=**0**;

sum\_deno\_delta\_tau=**0**;

end;

f\_tau=w\_pm\*(test\_LogOR-complet\_yw\_tau)\*\***2**;

deno\_delta\_tau=w\_pm\*\***2**\*(test\_LogOR-complet\_yw\_tau)\*\***2**;

\* Calculate terms that are summed across studies ...;

sum\_f\_tau+f\_tau;

sum\_deno\_delta\_tau+deno\_delta\_tau;

\* and sum them...;

if last.simruns then do;

a\_f\_tau=sum\_f\_tau-(&nstudies-**1**);

fin\_f\_tau=max(a\_f\_tau,**0**);

delta\_tau=max(**0**, (sum\_f\_tau-(&nstudies-**1**))/sum\_deno\_delta\_tau);

end;

run;

/\*proc print data=pm;run;\*/

\*determine the maximum of fin\_f\_tau across meta-analysis so that the loop is repeated until the stoping thresehold is reached in all meta-analysis;

proc means data=pm n;

var study;

output out=outmeanspm max(fin\_f\_tau)=max\_fin\_f\_tau;

run;

/\*proc print data=outmeanspm;\*/

\*assign the actuall max\_fin\_f\_tau as macro variable;

data pmeins;

set outmeanspm;

call symputx('max\_fin\_f\_tau', max\_fin\_f\_tau);

run;

%put &max\_fin\_f\_tau;

run;

data pm;

set pm;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_tau\_square\_pm\_temp1=**0**;

end;

tau\_square\_pm\_temp1=tau\_square\_pm+delta\_tau;

\* Calculate terms that are summed across studies ...;

sum\_tau\_square\_pm\_temp1+tau\_square\_pm\_temp1;

\* and sum them...;

if last.simruns then do;

tau\_square\_pm\_temp1=sum\_tau\_square\_pm\_temp1;

end;

run;

/\*proc print data=pm;\*/

\*imput missing column values of tau\_square\_pm which are necessary for summed terms in the next loop;

data pm;

do until(tau\_square\_pm\_temp1 or last);

set pm end=last;

if not missing(tau\_square\_pm\_temp1) then tau\_square\_pm\_neu=tau\_square\_pm\_temp1;

end;

do until(tau\_square\_pm\_temp1 or \_last);

set pm end=\_last;

output;

end;

run;

/\*proc print data=pm;\*/

\*drop variables that depend on tau\_square\_pm and conseqeuently change values in each loop;

data pm;

set pm;

by simruns;

tau\_square\_pm=tau\_square\_pm\_neu;

drop yw\_tau sum\_w\_pm sum\_num\_yw\_tau w\_pm num\_yw\_tau complet\_yw\_tau sum\_f\_tau sum\_deno\_delta\_tau f\_tau deno\_delta\_tau a\_f\_tau fin\_f\_tau delta\_tau sum\_tau\_square\_pm\_temp1 tau\_square\_pm\_temp1 tau\_square\_pm\_neu;

run;

/\*proc print data=pm;\*/

%end;

**%mend** PmEstimator;

options mlogic mprint symbolgen;

%***PmEstimator***(data=pmstart, start\_tau\_square\_pm=**0**, nstudies=nstudies, iterations=**1000**);

\*prepare data for the Paule and Mandel based REM;

**data** rempmeins;

set pm;

\* Compute the random effects weight for each study;

rem\_w\_pm=**1**/(vari+tau\_square\_pm);

**run**;

/\*proc print data=rempmeins;run;\*/

\* Compute the random effects estimator theta\_rem and its variance;

**data** rempm(keep=simruns nstudies LogOR\_IVRE\_PM var\_theta\_rem\_pm se\_theta\_rem\_pm);

set rempmeins end=lastrecord;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_rem\_times\_w\_pm=**0**;sum\_rem\_w\_pm=**0**;

end;

\* Calculate terms that are summed across studies ...;

test\_LogOR\_times\_rem\_w\_pm = rem\_w\_pm\*test\_LogOR;

\* ... and sum them;

sum\_rem\_w\_pm+rem\_w\_pm;

sum\_rem\_times\_w\_pm + test\_LogOR\_times\_rem\_w\_pm;

\* When the last observation is reached, calculate estimates and output them;

if last.simruns then do;

LogOR\_IVRE\_PM=sum\_rem\_times\_w\_pm/sum\_rem\_w\_pm;

var\_theta\_rem\_pm=**1**/sum\_rem\_w\_pm;

se\_theta\_rem\_pm=**1**/sqrt(sum\_rem\_w\_pm);

output;

end;

title"rempm";

**run**;

/\*proc print data=rempm;run;\*/

\* ... explode the data set rempm...;

**data** rempmexplode; set rempm; by simruns; do study=**1** to nstudies; output; end; **run**;

/\*proc print data=rempmexplode;run;\*/

\* ... and merge with the data set rempmeins;

**data** pmfinal;

merge rempmeins rempmexplode;

by simruns study;

**run**;

/\* proc print data=pmfinal;run;\*/

\*\*\*Compute the the Paule und Mandel heterogenity estimator based REM with normal confidence limits\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** LogOR\_PMRE(keep=simruns nstudies tau\_square\_pm LogOR\_PMRE var\_theta\_rem\_pm se\_theta\_rem\_pm var\_PMRE

CI95L\_LogOR\_PMRE CI95U\_LogOR\_PMRE se\_PMRE);

set pmfinal end=lastrecord;

LogOR\_PMRE=LogOR\_IVRE\_PM;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_rem\_w\_pm=**0**;

end;

\* ... and sum weigths;

sum\_rem\_w\_pm+rem\_w\_pm;

\* When the last observation is reached calculate estimates and output them;

if last.simruns then do;

var\_PMRE=**1**/sum\_rem\_w\_pm;

se\_PMRE=sqrt(var\_PMRE);

CI95L\_LogOR\_PMRE = LogOR\_PMRE - quantile('normal',**.975**)\* se\_PMRE;

CI95U\_LogOR\_PMRE = LogOR\_PMRE + quantile('normal',**.975**)\* se\_PMRE;

output;

end;

title"Paule und Mandel";

**run**;

/\*proc print data=LogOR\_PMRE; run;\*/

\*\*\*Compute the the Paule und Mandel heterogenity estimator based REM with (modified Hartung/Knapp Confidence limits)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** LogOR\_PMHK(keep=simruns nstudies tau\_square\_pm LogOR\_PMHK var\_theta\_rem\_pm se\_theta\_rem\_pm var\_pm\_hksj

se\_pm\_hksj CI95L\_LogOR\_PMHK CI95U\_LogOR\_PMHK);

set pmfinal end=lastrecord;

LogOR\_PMHK=LogOR\_IVRE\_PM;

by simruns;

\* Initialize terms that are summed up;

if first.simruns then do;

sum\_rem\_w\_pm=**0**;

end;

\* ... and sum weigths;

sum\_rem\_w\_pm+rem\_w\_pm;

\* When the last observation is reached calculate estimates and output them;

if last.simruns then do;

var\_pm\_hksj=**1**/sum\_rem\_w\_pm;

se\_pm\_hksj=sqrt(var\_pm\_hksj);

CI95L\_LogOR\_PMHK = LogOR\_PMHK - quantile('T',**.975**,nstudies-**1**)\* se\_pm\_hksj;

CI95U\_LogOR\_PMHK = LogOR\_PMHK + quantile('T',**.975**,nstudies-**1**)\* se\_pm\_hksj;

output;

end;

title"Paule/Mandel, Hartung/Knapp";

/\*proc print data=LogOR\_PMHK; run;\*/

**run**;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\* Beta-binomial model \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

ods select none;

**proc** **nlmixed** data=simdatadouble;

by simruns;

parms / bydata data=startingvalues\_Beta\_Binomial;

bounds **0.00001**<=rho<=**0.99999**;

mu= exp(b0 + b\_treat\*treatment)/ (**1** + exp(b0 + b\_treat\*treatment));

varianz=(**1**-rho)/rho;

\* alpha and beta are the parameters of the beta distribution;

alpha=mu\*(**1**-rho)/rho;

beta=(**1**-mu)\*(**1**-rho)/rho;

ll= lgamma(n+**1**)+lgamma(success+alpha)+lgamma(n-success+beta)+lgamma(alpha+beta)

-lgamma(success+**1**)-lgamma(n-success+**1**)-lgamma(n+alpha+beta) -lgamma(alpha)-lgamma(beta);

model success ~ general(ll);

\* Estimate the treatment effect;

estimate "LogOR" b\_treat;

ods output AdditionalEstimates=LogOR\_BBIN\_temp(where=(Label="LogOR") rename=(Estimate=LogOR\_BBIN StandardError=SE\_LogOR\_BBIN));

title"Beta-binomial model, Logit-Link for OR, NLMIXED";

**run**;

ods select all;

**data** LogOR\_BBIN(keep=simruns LogOR\_BBIN SE\_LogOR\_BBIN CI95L\_LogOR\_BBIN CI95U\_LogOR\_BBIN);

set LogOR\_BBIN\_temp;

CI95L\_LogOR\_BBIN = LogOR\_BBIN - probit(**0.975**)\* SE\_LogOR\_BBIN;

CI95U\_LogOR\_BBIN = LogOR\_BBIN + probit(**0.975**)\* SE\_LogOR\_BBIN;

**run**;

/\*proc print data=LogOR\_BBIN;run;\*/

\* Estimation of t-distibuted confidence limits with two times number of studies is the number of degress of freedom;

**data** LogOR\_BB2N(keep=simruns LogOR\_BB2N SE\_LogOR\_BB2N CI95L\_LogOR\_BB2N CI95U\_LogOR\_BB2N);

set LogOR\_BBIN\_temp;

LogOR\_BB2N=LogOR\_BBIN;

SE\_LogOR\_BB2N=SE\_LogOR\_BBIN;

CI95L\_LogOR\_BB2N = LogOR\_BB2N - quantile('T',**.975**,&nstudy\***2**)\* SE\_LogOR\_BB2N;

CI95U\_LogOR\_BB2N = LogOR\_BB2N + quantile('T',**.975**,&nstudy\***2**)\* SE\_LogOR\_BB2N;

**run**;

/\*proc print data=LogOR\_BB2N;run;\*/

\* Estimation of t-distibuted confidence limits with the number of studies is the number of degress of freedom;

**data** LogOR\_BB1N(keep=simruns LogOR\_BB1N SE\_LogOR\_BB1N CI95L\_LogOR\_BB1N CI95U\_LogOR\_BB1N);

set LogOR\_BBIN\_temp;

LogOR\_BB1N=LogOR\_BBIN;

SE\_LogOR\_BB1N=SE\_LogOR\_BBIN;

CI95L\_LogOR\_BB1N = LogOR\_BB1N - quantile('T',**.975**,&nstudy)\* SE\_LogOR\_BB1N;

CI95U\_LogOR\_BB1N = LogOR\_BB1N + quantile('T',**.975**,&nstudy)\* SE\_LogOR\_BB1N;

**run**;

/\*proc print data=LogOR\_BB1N;run;\*/

\* Estimation of t-distibuted confidence limits with the number observation minus three ist he number of degress of freedom;

**data** LogOR\_BBMP(keep=simruns LogOR\_BBMP SE\_LogOR\_BBMP CI95L\_LogOR\_BBMP CI95U\_LogOR\_BBMP);

set LogOR\_BBIN\_temp;

LogOR\_BBMP=LogOR\_BBIN;

SE\_LogOR\_BBMP=SE\_LogOR\_BBIN;

CI95L\_LogOR\_BBMP = LogOR\_BBMP - quantile('T',**.975**,&nstudy\***2**-**3**)\* SE\_LogOR\_BBMP;

CI95U\_LogOR\_BBMP = LogOR\_BBMP + quantile('T',**.975**,&nstudy\***2**-**3**)\* SE\_LogOR\_BBMP;

**run**;

/\*proc print data=LogOR\_BBMP;run;\*/

\* Estimation of the beta-binomialen models with PROC FMM;

ods select none;

**proc** **fmm** data=simdatadouble;

by simruns;

model success/n = treatment / dist=betabinomial link=logit cl;

ods output ParameterEstimates=LogOR\_BBFM\_temp(where=(Effect="treatment") rename=(Estimate=LogOR\_BBFM StdErr=SE\_LogOR\_BBFM));

title"Beta-binomial model, Logit-Link for OR, FMM";

**run**;

ods select all;

**data** LogOR\_BBFM(keep=simruns LogOR\_BBFM SE\_LogOR\_BBFM CI95L\_LogOR\_BBFM CI95U\_LogOR\_BBFM);

set LogOR\_BBFM\_temp;

CI95L\_LogOR\_BBFM = LogOR\_BBFM - probit(**0.975**)\* SE\_LogOR\_BBFM;

CI95U\_LogOR\_BBFM = LogOR\_BBFM + probit(**0.975**)\* SE\_LogOR\_BBFM;

**run**;

/\*proc print data=LogOR\_BBFM;run;\*/

\* Estimation of the beta-binomialen models with quasi-likelihood-estimation using PROC GLIMMIX;

ods select none;

**proc** **glimmix** data=simdatadouble order=formatted;

by simruns;

class treatment;

model success = treatment / s cl;

p= exp(\_linp\_)/(**1**+exp(\_linp\_));

\_mu\_=n\*p;

\_variance\_= n\*p\*(**1**-p)\*(**1**+(n-**1**)\*\_phi\_);

format treatment treatf.;

ods output ParameterEstimates=LogOR\_BBQL\_temp(where=(treatment=**1**) rename=(Estimate=LogOR\_BBQL StdErr=SE\_LogOR\_BBQL));

title'Quasi-Likelihood-Schätzung eines Beta-Binomial-Modells, PROC GLIMMIX';

**run**;

ods select all;

**data** LogOR\_BBQL(keep=simruns LogOR\_BBQL SE\_LogOR\_BBQL CI95L\_LogOR\_BBQL CI95U\_LogOR\_BBQL);

set LogOR\_BBQL\_temp;

CI95L\_LogOR\_BBQL = LogOR\_BBQL - probit(**0.975**)\* SE\_LogOR\_BBQL;

CI95U\_LogOR\_BBQL = LogOR\_BBQL + probit(**0.975**)\* SE\_LogOR\_BBQL;

**run**;

/\*proc print data=LogOR\_BBQL;run;\*/

**data** LogOR\_BBQD(keep=simruns LogOR\_BBQD SE\_LogOR\_BBQD CI95L\_LogOR\_BBQD CI95U\_LogOR\_BBQD);

set LogOR\_BBQL\_temp;

LogOR\_BBQD=LogOR\_BBQL;

SE\_LogOR\_BBQD=SE\_LogOR\_BBQL;

CI95L\_LogOR\_BBQD = LogOR\_BBQD - quantile('T',**.975**,&nstudy\***2**-**2**)\* SE\_LogOR\_BBQD;

CI95U\_LogOR\_BBQD = LogOR\_BBQD + quantile('T',**.975**,&nstudy\***2**-**2**)\* SE\_LogOR\_BBQD;

**run**;

/\*proc print data=LogOR\_BBQD;run;\*/

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\* SUMMARIZE SIMULATION RESULTS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

**data** result\_&ident;

\* Collect information for this simulation run;

Ident="&ident";

\* Merge result data sets from the different procedures;

merge LogOR\_MaHa yusufpeto LogOR\_DLRE LogOR\_HKSJ LogOR\_PMRE LogOR\_PMHK LogOR\_BBIN LogOR\_BB2N LogOR\_BB1N LogOR\_BBMP LogOR\_BBFM LogOR\_BBQL LogOR\_BBQD

;

by simruns;

\* True values;

true\_OR=&true\_OR;

true\_LogOR=log(true\_OR);

\* Macro to calculate bias and 95% coverage;

**%macro** calcresp(method,true,nullvalue);

bias\_&method=&method-true\_&true;

coverage\_&method=**.**;

if CI95L\_&method<true\_&true and CI95U\_&method>true\_&true then coverage\_&method=**1**;

if CI95L\_&method>true\_&true or CI95U\_&method<true\_&true then coverage\_&method=**0**;

if &method=**.** or CI95L\_&method=**.** or CI95U\_&method=**.** then coverage\_&method=**.**;

power\_&method=**.**;

if CI95L\_&method>&nullvalue or CI95U\_&method<&nullvalue then power\_&method=**1**;

if CI95L\_&method<&nullvalue and CI95U\_&method>&nullvalue then power\_&method=**0**;

if &method=**.** or CI95L\_&method=**.** or CI95U\_&method=**.** then power\_&method=**.**;

**%mend** calcresp;

%***calcresp***(LogOR\_MaHa,LogOR,**0**);

%***calcresp***(LogOR\_YPET,LogOR,**0**);

%***calcresp***(LogOR\_DLRE,LogOR,**0**);

%***calcresp***(LogOR\_HKSJ,LogOR,**0**);

%***calcresp***(LogOR\_mKH\_,LogOR,**0**);

%***calcresp***(LogOR\_PMRE,LogOR,**0**);

%***calcresp***(LogOR\_PMHK,LogOR,**0**);

%***calcresp***(LogOR\_BBIN,LogOR,**0**); %***calcresp***(LogOR\_BB2N,LogOR,**0**); %***calcresp***(LogOR\_BB1N,LogOR,**0**); %***calcresp***(LogOR\_BBMP,LogOR,**0**);

%***calcresp***(LogOR\_BBFM,LogOR,**0**); %***calcresp***(LogOR\_BBQL,LogOR,**0**); %***calcresp***(LogOR\_BBQD,LogOR,**0**);

**run**;

/\*

proc contents data=result\_&ident position;run;

proc print data=result\_&ident;

run;

\*/

/\*proc print data=result\_&ident;

var LogOR\_MaHa LogOR\_YPET LogOR\_DLRE LogOR\_HKSJ LogOR\_mKH\_ LogOR\_PMRE LogOR\_PMHK

LogOR\_BBIN LogOR\_BB2N LogOR\_BB1N LogOR\_BBMP LogOR\_BBFM LogOR\_BBQL LogOR\_BBQD;

title"Display Estimates";

run;\*/

/\*proc means data=result\_&ident n nmiss mean median min max maxdec=3;

var LogOR\_MaHa LogOR\_YPET LogOR\_DLRE LogOR\_HKSJ LogOR\_mKH\_ LogOR\_PMRE LogOR\_PMHK

LogOR\_BBIN LogOR\_BB2N LogOR\_BB1N LogOR\_BBMP LogOR\_BBFM LogOR\_BBQL LogOR\_BBQD;

title"Analyse all Estimates";

run;\*/

/\*

proc print data=result\_&ident;

var bias\_LogOR\_MaHa bias\_LogOR\_YPET bias\_LogOR\_DLRE bias\_LogOR\_KKSJ bias\_LogOR\_mKH\_ bias\_LogOR\_PMRE bias\_LogOR\_PMHK

bias\_LogOR\_BBIN bias\_LogOR\_BB2N bias\_LogOR\_BB1N bias\_LogOR\_BBMP bias\_LogOR\_BBFM bias\_LogOR\_BBQL bias\_LogOR\_BBQD;

title"Display Biases";

run;

\*/

/\*proc means data=result\_&ident n nmiss mean median min max maxdec=3;

var bias\_LogOR\_MaHa bias\_LogOR\_YPET bias\_LogOR\_DLRE bias\_LogOR\_HKSJ bias\_LogOR\_mKH\_ bias\_LogOR\_PMRE bias\_LogOR\_PMHK

bias\_LogOR\_BBIN bias\_LogOR\_BB2N bias\_LogOR\_BB1N bias\_LogOR\_BBMP bias\_LogOR\_BBFM bias\_LogOR\_BBQL bias\_LogOR\_BBQD;

title"Analyse all Biases";

run;\*/

/\*

proc print data=result\_&ident;

var coverage\_LogOR\_MaHa coverage\_LogOR\_YPET coverage\_LogOR\_DLRE coverage\_LogOR\_HKSJ coverage\_LogOR\_mKH\_ coverage\_LogOR\_PMRE coverage\_LogOR\_PMHK

coverage\_LogOR\_BBIN coverage\_LogOR\_BB2N coverage\_LogOR\_BB1N coverage\_LogOR\_BBMP coverage\_LogOR\_BBFM coverage\_LogOR\_BBQL coverage\_LogOR\_BBQD;

title"Display Coverages";

run;

\*/

/\*proc means data=result\_&ident n nmiss mean maxdec=3;

var coverage\_LogOR\_MaHa coverage\_LogOR\_YPET coverage\_LogOR\_DLRE coverage\_LogOR\_HKSJ coverage\_LogOR\_mKH\_ coverage\_LogOR\_PMRE coverage\_LogOR\_PMHK

coverage\_LogOR\_BBIN coverage\_LogOR\_BB2N coverage\_LogOR\_BB1N coverage\_LogOR\_BBMP coverage\_LogOR\_BBFM coverage\_LogOR\_BBQL coverage\_LogOR\_BBQD;

title"Analyse all Coverages";

run;\*/

/\* Delete data sets \*/

**proc** **datasets**;

delete CHECK\_DOUBLEZERO CHECK\_MA\_WITHOUT\_EVENTS COV\_LOGIT COV\_LOGIT\_TEMP IDENT KENDALLCORR LOGOR\_BBIN LOGOR\_BBIN\_TEMP LOGOR\_CLRW LOGOR\_CLRW\_TEMP

LOGOR\_COLL LOGOR\_MAHA LOGRR\_BBIN LOGRR\_BBIN\_TEMP LOGRR\_COLL LOGRR\_FREQ LOGRR\_MAHA LOGRR\_MAHA\_TEMP MA\_WITHOUT\_EVENTS MEANSOUT\_CHECK\_DOUBLEZERO

MEANSOUT\_CHECK\_MA\_WITHOUT\_EVENTS MEANSOUT\_MA\_WITHOUT\_EVENTS MEANSOUT\_OVERALL\_P MEAN\_START\_CAI\_RBRR\_TEMP OR\_COLL\_TEMP OR\_FREQ\_TEMP OR\_MAHA\_TEMP

PC\_GAMMA PC\_GAMMA\_TEMP1 PEARSONCORR RD\_BBIN RD\_BBIN\_TEMP RD\_COLL RR\_COLL\_TEMP RR\_FREQ\_TEMP RR\_MAHA\_TEMP SIMDATA SIMDATADOUBLE SIMDATADOUBLE\_TEMP1

SIMDATAEXPLODE SIMPLESTATS\_LOG\_PC SIMPLESTATS\_LOG\_PC\_TEMP SIMPLESTATS\_LOG\_PT SIMPLESTATS\_LOG\_PT\_TEMP SIMPLESTATS\_PC SIMPLESTATS\_PT SPEARMANCORR

STARTINGVALUES\_BETA\_BINOMIAL STARTINGVALUES\_CAI\_RBFR STARTINGVALUES\_CAI\_RBRR STARTINGVALUES\_CAI\_RBRR\_TEMP STARTINGVALUES\_STIJNEN\_BN YUSUFPETO

LOGOR\_DLRE LOGOR\_HKSJ LOGOR\_PMRE LOGOR\_PMHK IVRE QCALCIVRE QCALCIVREEXPLODE REMEINS REM REMEXPLODE DLEINS DLHKSJ OUTMEANS OUTGLM PM PMSTART PMEINS

PMIVRE REMPMEINS REMPMEXPLODE REMPM PMFINAL OUTMEANSPM;

**run**;**quit**;

**%mend** FewStudiesSim;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\* RUN THE SIMULATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

options mlogic mprint;

\* REM;

%***FewStudiesSim***(nstudy=**50**, true\_or=**0.684**, rem=**1**, large\_study=**1**, seedsim=**2534786**,nsimruns=**10000**, printsimdata=**0**);

/\*proc print data=result\_h0feme02; run;\*/

libname results "H:\Biometrie Studium\Masterarbeit\Simulationen FEWSTUDIES\Results";

**data** results.result\_h1reml50;

set result\_h1reml50;

**run**;